

FIG. 1

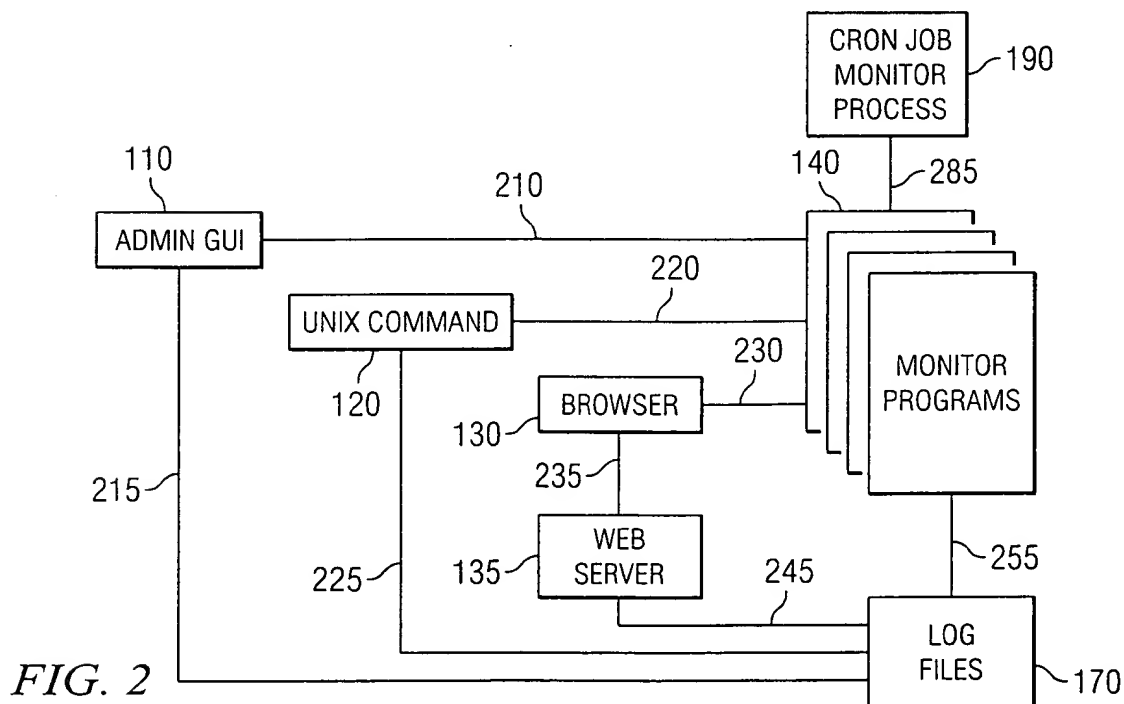
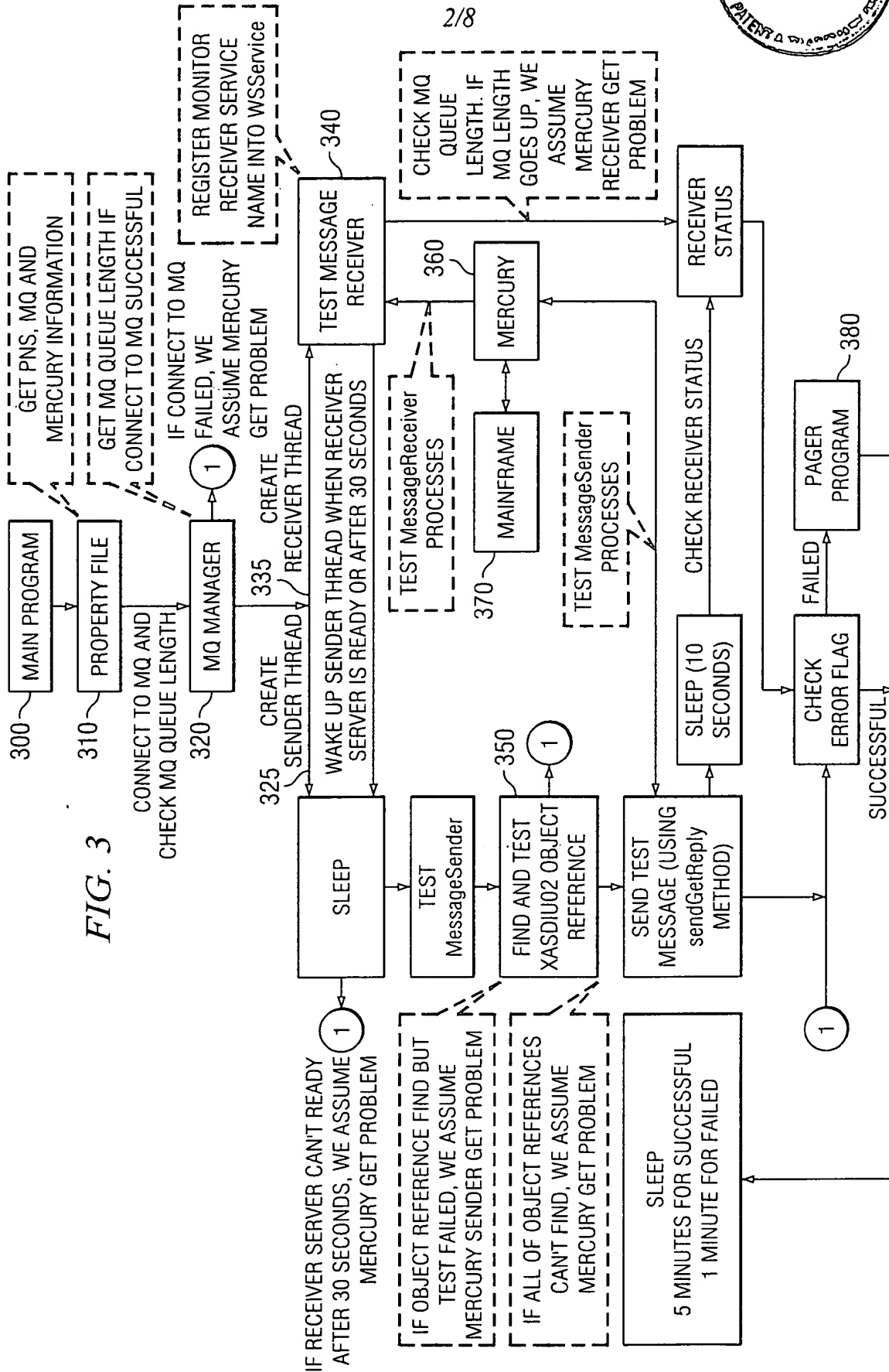


FIG. 2



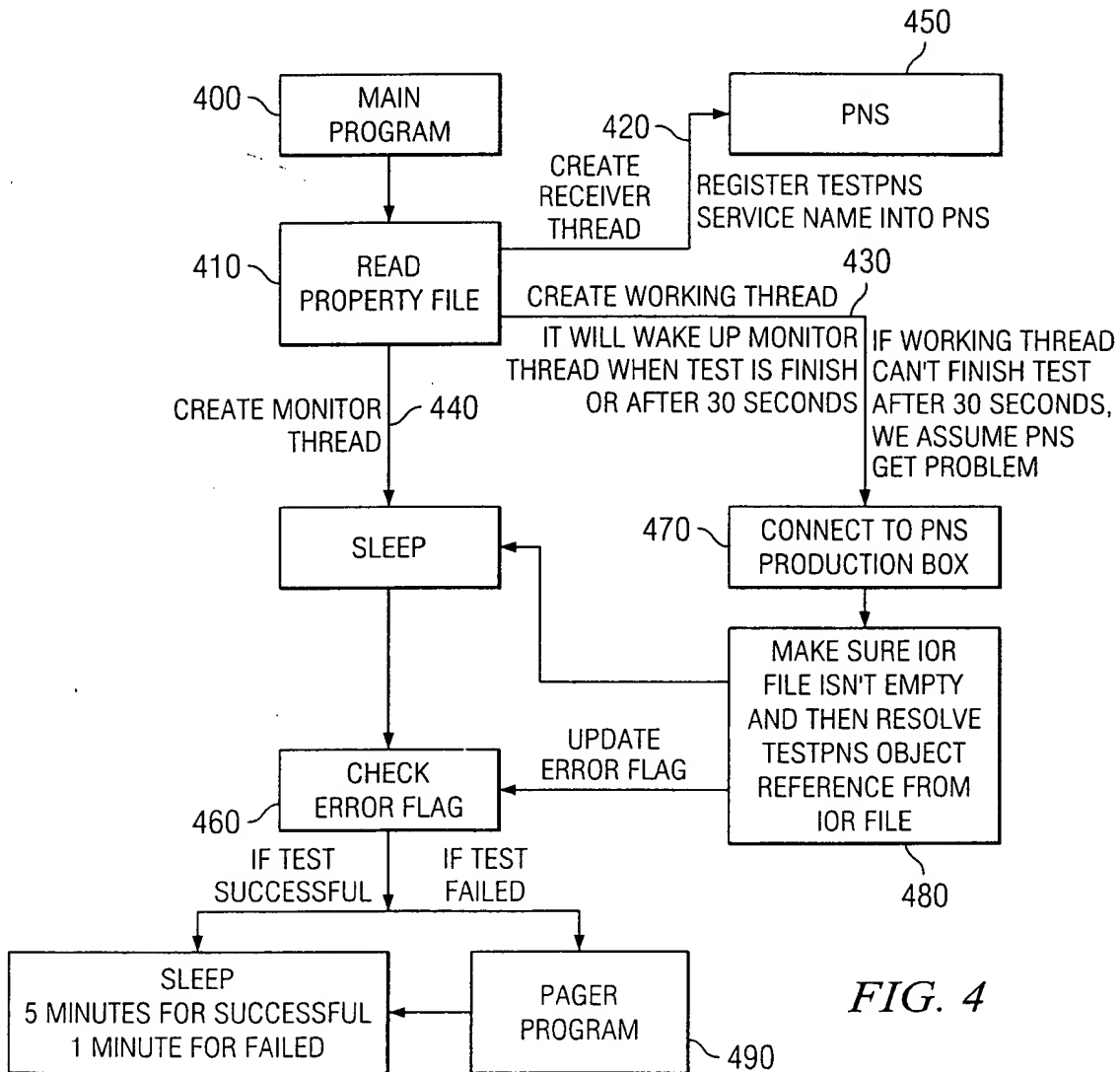
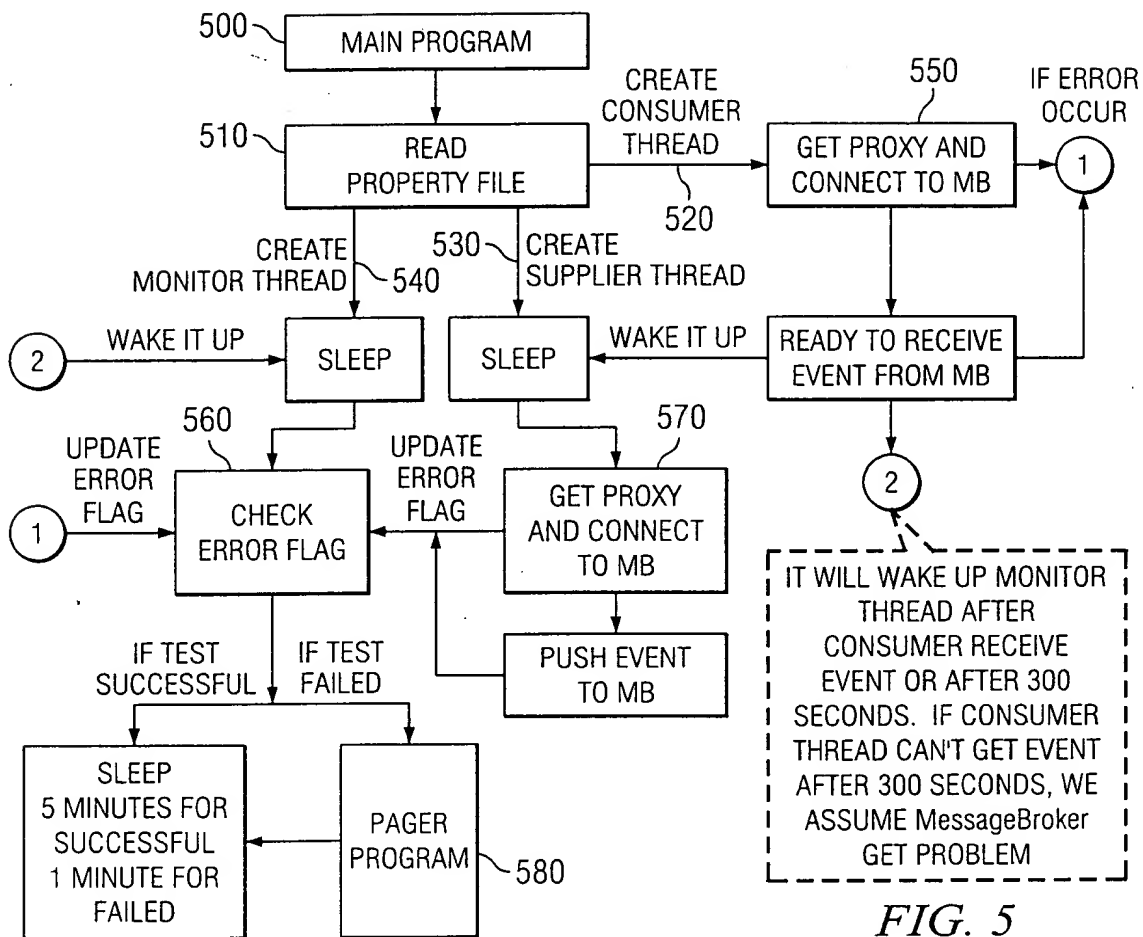


FIG. 4





5/8

FIG. 6

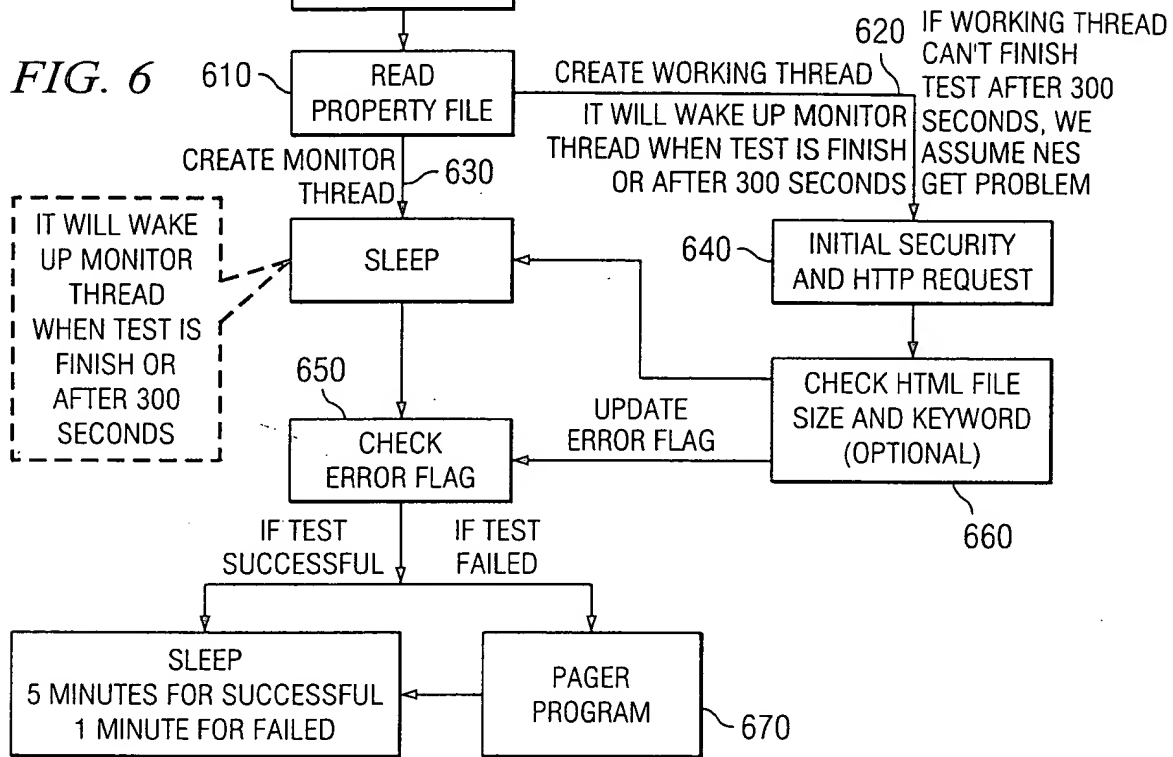
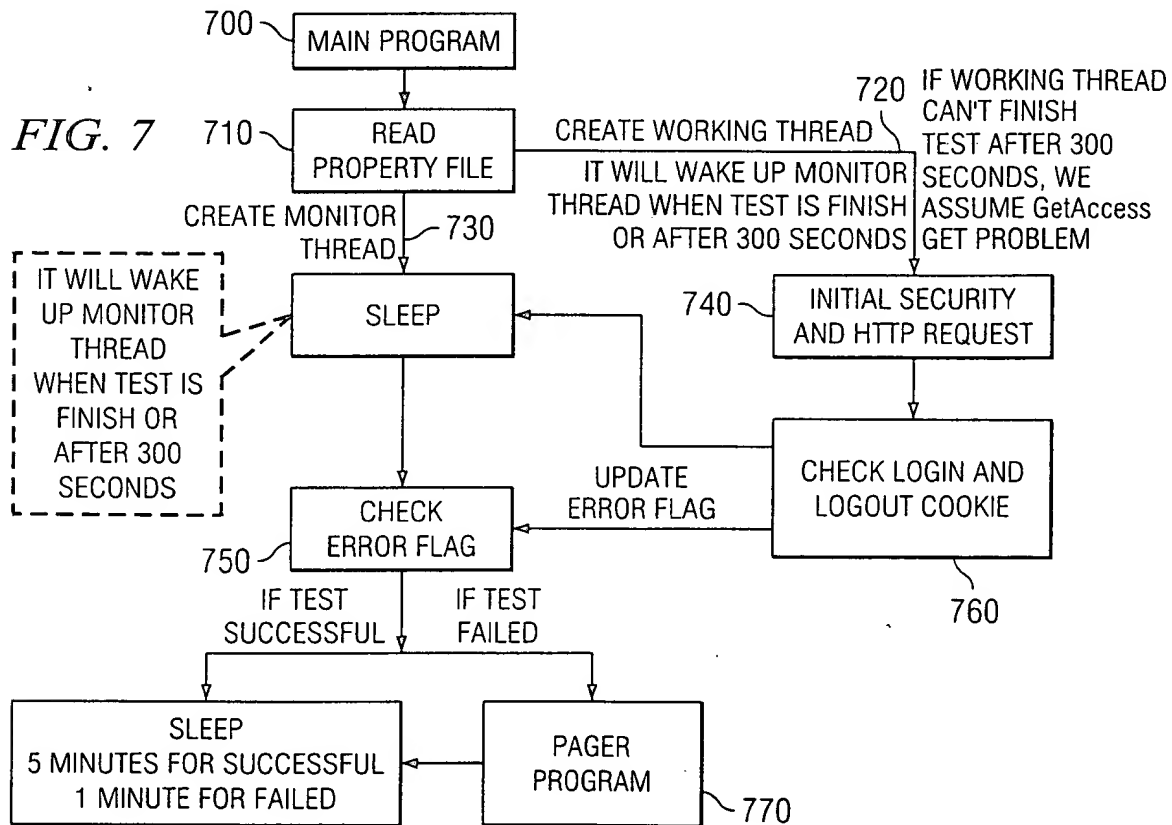


FIG. 7



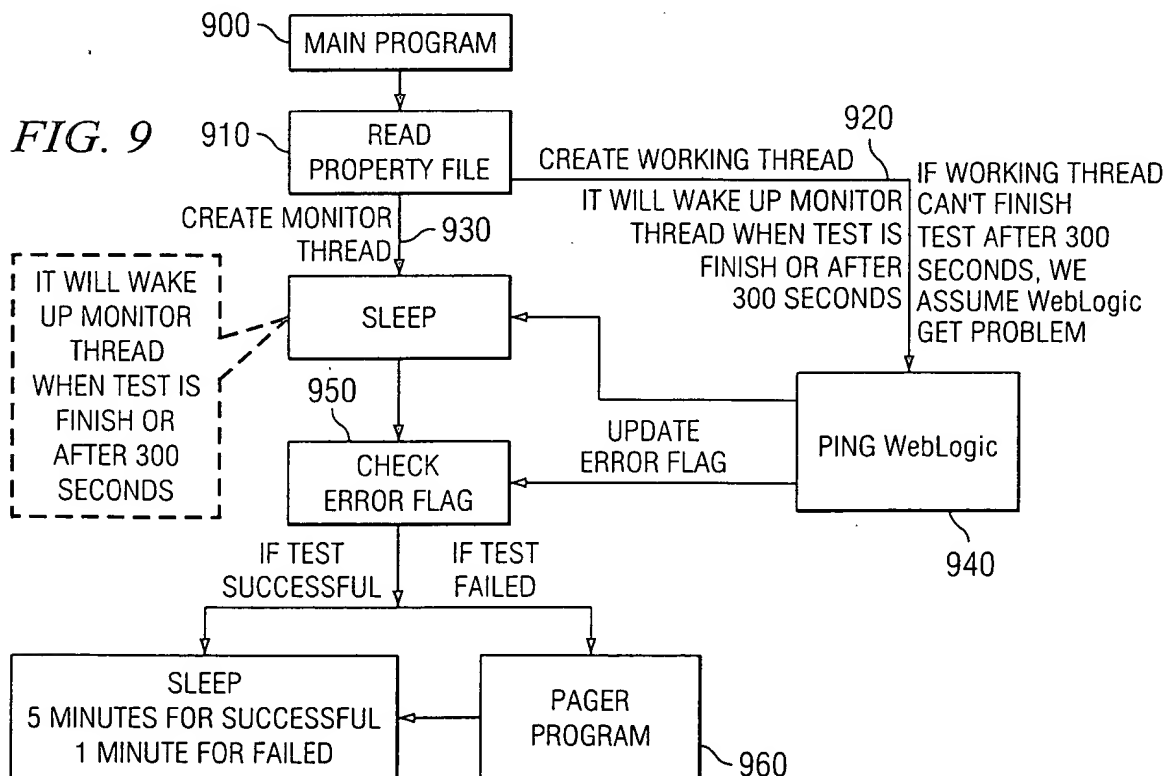
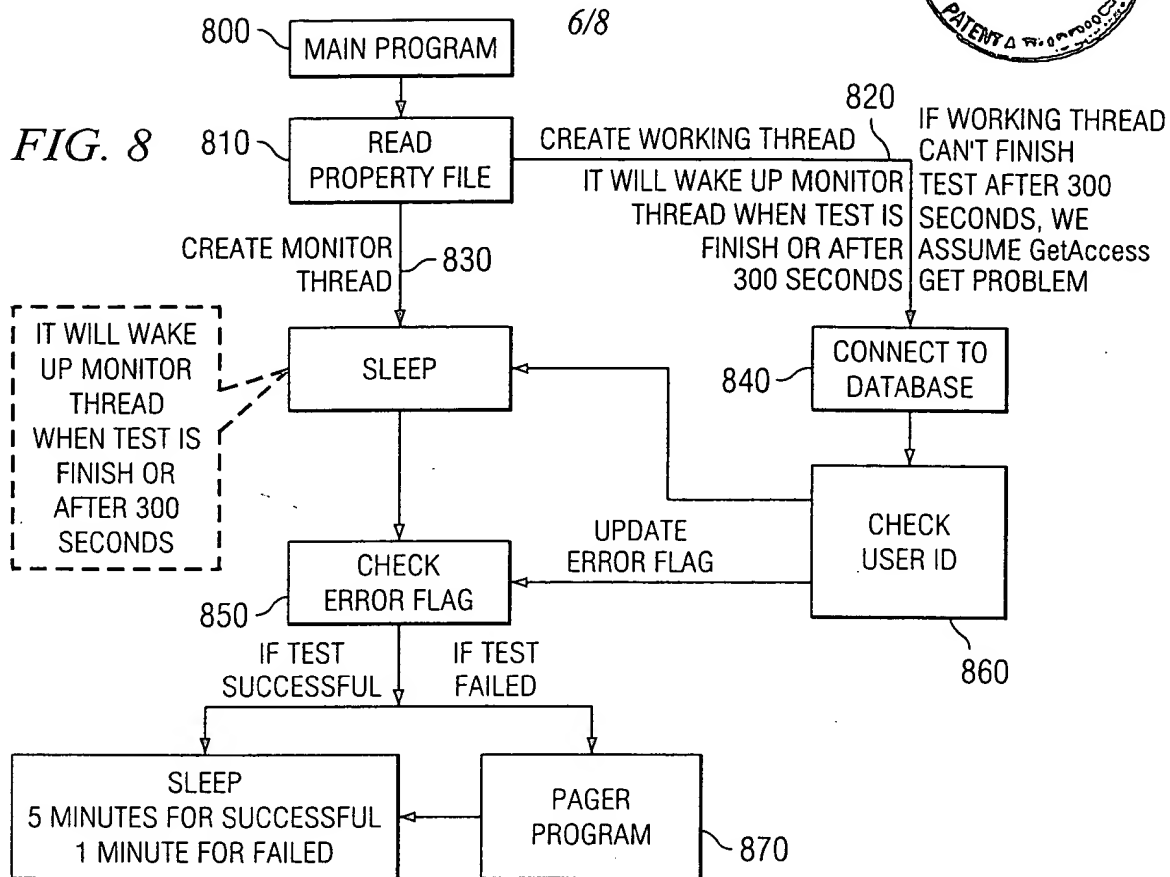
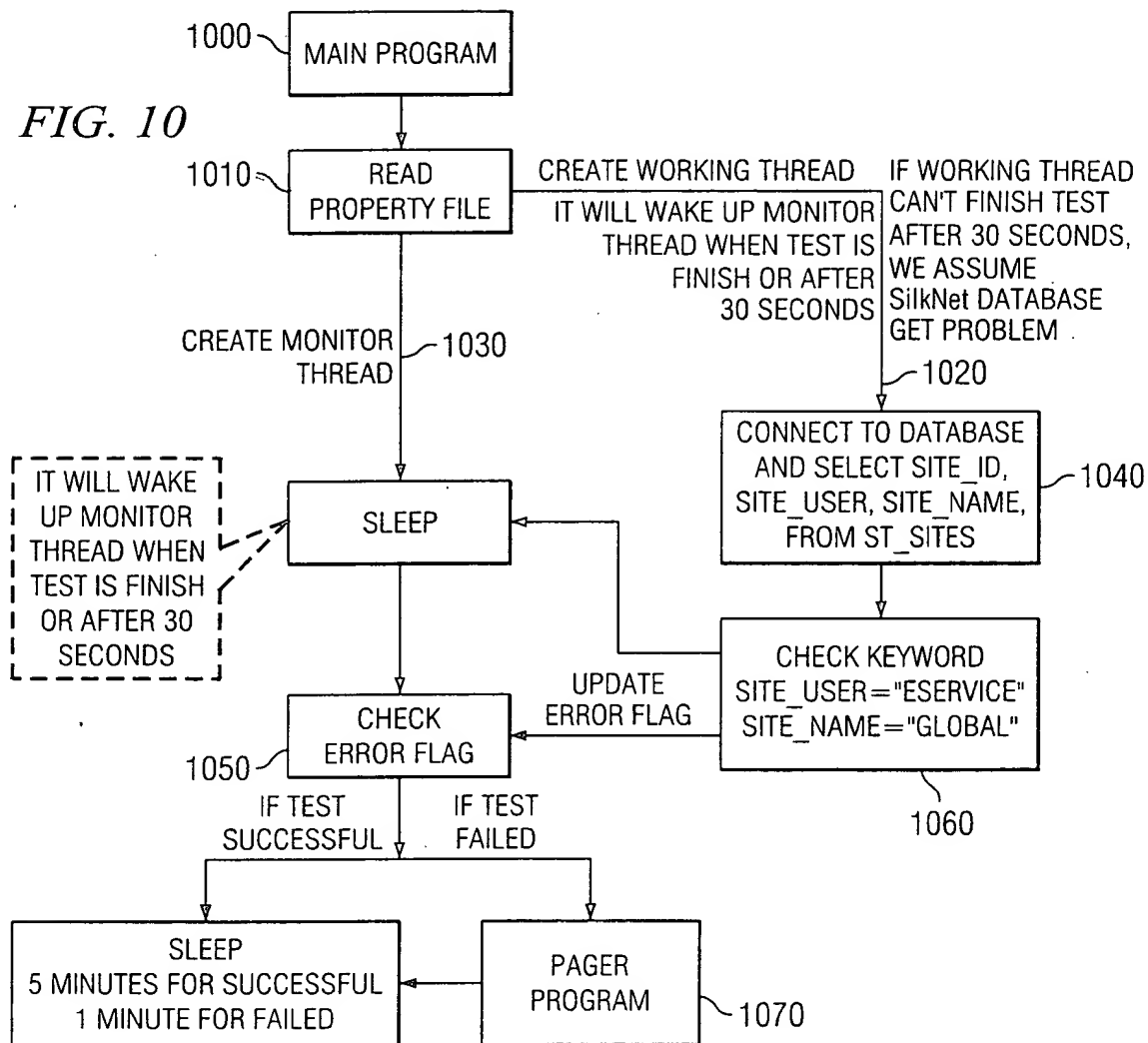




FIG. 10



**FIG. 11**

```
graph TD
    1100[MAIN PROGRAM] --> 1110[READ PROPERTY FILE]
    1110 --> 1130[CREATE MONITOR THREAD]
    1130 --> 1140[SLEEP]
    1140 --> 1150[CHECK ERROR FLAG]
    1150 -- "IF TEST SUCCESSFUL" --> 1160[SLEEP 5 MINUTES FOR SUCCESSFUL  
1 MINUTE FOR FAILED]
    1150 -- "IF TEST FAILED" --> 1170[CONNECT TO DAEMON SERVER TO TEST TRANSACTION]
    1170 --> 1180[PAGER PROGRAM]
    1180 --> 1160
    1170 -- "UPDATE ERROR FLAG. IF TRANSACTION THREAD CAN'T FINISH TEST AFTER 30 SECONDS, WE ASSUME DAEMON SERVER GET PROBLEM" --> 1150
    1170 -- "TRY TO FIND AND TEST TPBROKER PROCESSES" --> 1120[CREATE WORKING THREAD. IF WORKING THREAD CAN'T FINISH TEST AFTER 30 SECONDS, ASSUME TPBROKER PROCESSES GET PROBLEM]
    1120 -- "RETURN SUCCESSFUL OR FAILED" --> 1110
```

The flowchart illustrates the TPBROKER process. It begins with the MAIN PROGRAM (1100) which reads a PROPERTY FILE (1110). A MONITOR THREAD is created (1130) and enters a SLEEP state (1140). A dashed box notes: "IT WILL WAKE UP MONITOR THREAD WHEN TEST IS FINISH OR AFTER 30 SECONDS". The process then checks an ERROR FLAG (1150). If the test is successful, it sleeps for 5 minutes (or 1 minute if failed). If the test fails, it connects to a DAEMON SERVER (1170) to test the transaction. From the daemon server, it either updates the error flag (if the transaction thread can't finish the test after 30 seconds, assuming the daemon server got a problem) or tries to find and test TPBROKER processes. The latter path leads to creating a working thread (1120), which returns successful or failed to the monitor thread (1110). The process then loops back to the SLEEP state (1140) via the PAGER PROGRAM (1180).

